Sets For Foundational Representations?

A Design Case Study With Probability And Distributions

Peter C-H. Cheng[0000-0002-0355-5955]

Department of Informatics, University of Sussex, Brighton, BN1 9QJ, UK
p.c.h.cheng@sussex.ac.uk

Abstract. Ideas about sets are foundational to our understanding of many knowledge domains. And cognitive science tells us that the representation (notation or visualization) we use to encode the knowledge of a domain substantially determines what we can think and how easily we can reason about that domain. Therefore, how a representation encodes ideas about sets may substantially determine how readily we can comprehend, solve problems and learn about its domain. So, how should we design representations for knowledge rich domains to ensure that concepts about sets are readily accessible and also effectively integrated with the domain's other concepts? A case study is presented in which a representation for sets (Set Space Diagrams) is taken as a foundation for a representation for probability theory (Probability Space Diagrams) and then further extended as a representation for statistical distributions (Distribution Space Diagrams). Together the three representations constitute a unified framework that conceptually integrates knowledge across the three domains.

Keywords: Sets, probability, statistical distributions, notations, visualization, diagrams, knowledge recodification.

1 Introduction

Ideas about sets play a fundamental role in the knowledge of very many domains. If a topic includes notions about categories, groupings, classes, divisions, assemblies, partitions and so forth, sets are relevant to the comprehension and interpretation of that topic. Bakery production managers are thinking about sets when they assemble different products into shipments to fulfil a supermarket order. Biologists deal with sets when they attempt to classify the discovery of yet another new type of beetle in to the taxonomic hierarchy of insect species. You make set-based decisions as you type queries in to your favourite search engine. We hope that our doctor understands the implications of the full range of possible outcomes, true and false, positive and negative, when advising us about the results of a test for a life-threatening condition. A statistician considering whether values in two samples come from the same population is yet another example. Even pondering whether examples of this kind are boundless adds another example to the set.

Cognitive science has established that the representation (notation or visualization) we use for solving a problem fundamentally affects not just how easily we can solve the problem [14] but what we can understand and discover [19] and also what and how we learn [1]. For instance, alternative representations for the same problem, but which are isomorphic in structure, can make problem solving an order of magnitude more demanding [14].

Link the ideas of the previous two paragraphs together: In the design of a representation for a given domain, will how we choose to encode information about *sets* substantially impact our thinking and learning about that domain? In the *Representational Epistemic* method for representation design [1, 2], I argue that notation and visualization design should begin with an analysis of the fundamental conceptual structure of the target topic. In particular, we should identify the underpinning concepts that permeate all ideas in the domain: concepts which make the domain that domain rather than some other. That conceptual structure should be the basis for the design of representations. In many domains, sets constitute a major conceptual foundation, so should an effective representation closely integrate set information with other core concepts of the domain? Put another way, given the importance and ubiquity of sets, knowing how to effectively encode information about sets could comprise a general approach to enhance higher cognition across diverse subject areas.

This sixth workshop dedicated to Euler diagrams and the visualizations of sets attests to the importance of the question of how to represent sets. Diverse representations have been examined in these workshops and the literature more widely. Euler diagrams [8] and Venn diagrams [18] are, of course, classic graphical notations for sets, which rely upon degrees of spatial containment as their primary means of encoding relations. Node-link graphs can serve as representations of sets; Venn and Euler diagrams can be systematically redrawn as trees. These notations have been extended and formalized in order to serve as general graphical languages for the specification of complex systems (e.g., Spider diagrams [12]). Other notations have been developed that rely less on 2D spatial arrangements, including: *Linear Diagrams* [7]; Set Space Diagrams [10], (later also called *Linear Diagrams* [11]), which originated in *Probability Space Diagrams* [2]; *Categorical Pattern Diagrams* [3]. The development of all these graphical forms of set representations is, in part, a reaction to the inherent difficulties of reasoning with linear symbolic formulations of set theory.

Moving beyond expressing ideas purely about sets, we see many representations for knowledge rich domains that adopt (a) degrees of spatial containment or (b) graphs as devices to encode set based ideas. In Mendeleev's periodic table of the elements the columns and rows stand for chemical groups and periods, respectively. Taxonomies in biology and beyond are often shown as trees. Both devices were used in two systems I designed using the Representation Epistemic approach. In the *ROLLOUT* system for bakery production management and scheduling, rectangles standing for products are assembled into blocks that comprise shipments [5]. In *STARK-Exam*, a system for the problem of university examination time tabling, four nested layers of rectangles capture information about the allocation of students to exams, exams to rooms in particular time and daily slots, and interconnecting lines (edges) define preference constraints among components across the levels [4]. Jones [13] investigated whether Euler diagrams could

be used to specify search queries but found they were less effective than expressions with symbolic connectives. Spiegelhalter and colleagues [17] examined visualizations of risks about the future – many of which use the two devices – and concluded that many alternative representations are required to satisfy the needs of users with different levels of domain expertise and sophistication in graphical displays.

All this suggests that designing representations that coherently integrate set information with other information for knowledge rich domains will be a valuable activity. However, such a task will be challenging, because many set related concepts will likely need to be encoded. These include: elements within a group; empty sets and universal sets; relations among subsets, such as complements, union and intersection, within a trial; relations between trials, such as joint or disjointedness of subsets, or conditionals, which may be consistent or contrary to the order of occurrence of the trials; combinations of two (or more) trials with sets formed by Cartesian products; sophisticated relations such as De Morgan's laws. How can a representation of sets encode most of these ideas and yet be compatible with all the other ideas of a given domain? Further, is it possible to create a methodology for the design of such representations for any domain in general? This could be an interesting, and major, challenge for the *SetVR* and *Diagrams* research community to address.

The aim of this paper is to demonstrate the potential value of designing representations for complex domains starting with sets. Specifically, we will consider how representations that unifies *probability theory* and *statistical distributions* can be created by acknowledging the fundamental role of sets in both of those topics and then finding a graphical format for sets that extends naturally to probability and then extends naturally to distributions. The next section introduces the *Diagrammatic Statistics Framework*, DS Framework, which encompasses the three representations. Then to begin to demonstrate the coherence and efficacy of representations in the DS Framework, the subsequent three sections presents a detailed example in each domain. The final section draws out the wider implications of the design of the set-based representations.

2 Diagrammatic Statistics Framework

Fig. 1 shows the basic format of the representations for sets, probability and distributions. These are: *Set Space Diagrams*, SSDs, Fig. 1A; *Probability Space Diagrams*, PSDs, Fig. 1B; *Distribution Space Diagrams*, DSDs, Fig. 1C. The left side of Fig. 1 show situations with one trial and the right shows situations involving two trials.

A SSD defines a space within which labelled line segments (*lines* for short) are drawn to represent sets. Labels denotes the names or properties of the sets. Clusters of lines at the same altitude stand for a particular trial (universal set), Fig. 1Ai. Clusters at different altitudes represent different trials, Fig. 1Aii. The width of lines is arbitrary but their horizontal arrangement relative to each other encodes set relations (A_1=black, A_2=large). Under the clusters of lines in Fig. 1a, examples of elements with different properties are displayed to make concrete the relations among the sets. For example, the overlap of two lines is an intersection of their sets and the full horizontal extent of several lines represents the union of all the sets. A line split into parts horizontally with

the same label represents a single subset with that property (e.g., Fig. 1Aii, two lines of B_1). This permits the expression of complex relations among many subsets.

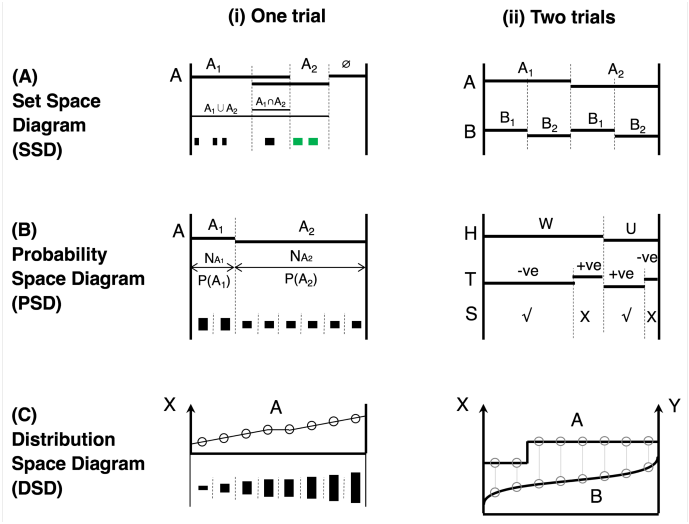

Fig. 1. Diagrammatic Statistics Framework

To compute the Cartesian product of two trials, all the lines of one set are drawn under each and every line of the other trial, Fig. 1Aii. Each column is a combination (e.g., A_1B_1). Complex contingencies among sets are expressed by omitting or adding lines to either of the two trials as needed (not shown).

Probability theory builds upon set theory, so the SSDs incorporates information about quantities of chance by imposing a metric interpretation of the length of each line, Fig. 1Bi. The bottom half of Fig. 1Bi shows the equi-space arrangements of different numbers of elements of two alternative properties, *A1* and *A2*. The length of the lines may be read as either the number of elements in the set or the probability of sampling elements with the particular property in one trial, depending on one's overall interpretation of the diagram. Further, the ratio of the length of line A_1 to A_2, in Fig. 1Bi, is a measure of the odds, $A_1{:}A_2$. Probabilities of relations among conditions can be read from the diagrams; e.g., A_1 and A_2 are disjoint so P(A_1 & A_2)=0 and P(A_1 or A_2)=1, in Fig. 1Bi.

Situations with more than one trial similarly enforce a metric interpretation of SSD line lengths to give PSDs for multi-trial situations. Fig. 1Bii shows the typical PSD model for situations involving tests. Trial *H* gives the base rates of some cases, such as a person's health status: well (*W*) or unwell (*U*). Trial *T* applies a test, which may give a positive (*+ve*) or a negative *(-ve*) result. The third lines give the status, *S*, of the test in relation to whether the person was genuinely healthy or not, which may be a correct (√) or incorrect (*X*) diagnosis. A doctor could use this PSD to discuss the likelihood that the diagnosis is in error, given the prevalence of the illness in the population.

Statistical distributions build upon probability theory, with the idea that each element is an equally likely outcome of a trial, and that multiple elements that are the same according on some measure will appear more frequently in a sample. So, to encode the idea of measuring a property of the elements of the population (total set), DSDs use (a) the vertical axis of the space to represent magnitudes of the property and (b) arrange the elements in rank order of those magnitudes. In Fig. 1Ci, measures of property *X* of trial *A* are shown, which happens to include two elements with the same value. This method of encoding magnitudes gives common distributions specific shapes in DSDs. For instance: uniformly distributed values are linear with a positive slope (e.g., left and right parts of curve A in Fig. 1Ci); constant values produce a straight flat line (middle of part of A in Fig. 1Ci); normal distributions have a rotational symmetric 'S' shape (Fig. 1Cii, curve B); skewed normal distributions distort the tails of the normal curve and kurtotic distributions either flatten the middle of the curve or exaggerate its slope. Measures of the centre and dispersion of distributions are encoded by simple spatial and geometric relations; for instance, the median is found by a simple bisection of the space, and quartiles by bisections of those bisections.

Together the SSDs, PSDs and DSDs representations constitute the DS Framework. SSDs encode information about sets using simple graphical constraints based on a 2D space where horizontal and vertical dimensions serve different representational functions. Recognizing that many concepts in the domain of probability are basically ideas grounded in notions about sets, the design of PSDs extended SSDs by simply placing an additional interpretive constraint on SSDs; specifically, the metric interpretation of horizontal line length as magnitudes of probability. Recognizing the many concepts in the nature of statistical distributions depend on ideas about probability, the design of DSDs extend PSDs (and SSDs) by adding additional constraints, with the interpretation of vertical space as magnitudes of domain variables and ordering data points (elements) by magnitude.

To demonstrate the potential of each representation the follow three sections consider selected examples in detail.

3 SSDs: Comparison to Euler Diagram

To form a solid foundation on which to build representations for the knowledge rich domains of probability and statistical distributions, the representation must be an effective representation for sets, in general. Among other things, such a representation must: support different forms of problem solving involving sets; enable displays of different configurations of sets to be straightforwardly drawn; allow such configurations to be readily distinguish from each other. The author has applied SSDs to a variety of problems, for example: SSDs can be used to demonstrate the validity of theorems, such as de Morgan's laws; they may be used to find formulas that summarize a given list of subsets; a variant of SSDs can be used to perform syllogistic inferences. (These will be described elsewhere.) SSDs appear to satisfy the first of the three requirements. The

rest of this section will compare SSDs to Euler diagrams, because they are widely considered to be effective visualizations and are used as foundational components for various representations systems (e.g., [12]).

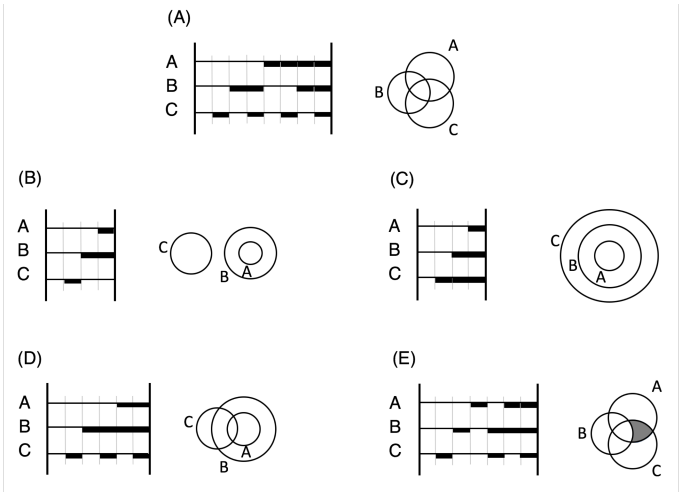

Fig. 2. Comparison of SSDs and Euler diagrams. (A) All subsets – Venn Diagram. (B) Sets C, B, AB. (C) Sets C, BC, ABC. (D) Sets C, B, BC, AB, ABC. (E) Sets C, B, A, BC, AB, ABC.

Euler diagrams are an alternative to Venn Diagrams for the presentation of information about sets and sometimes simpler to interpret and construct, because they omit zones that are empty. Fig. 2 shows selected Euler diagrams and corresponding SSDs. The caption lists the (non-empty) subsets displayed by each pair of diagrams. The thick lines in the SSDs stand for subsets with members and the thin lines stand for those that are empty. Fig. 2A shows diagrams for three sets, in which all possible subsets have members; the Euler diagram is (almost) a Venn diagram. This SSD is created by repeating the Cartesian product process that generated Fig. 1Aii, but twice over for *B*'s two pairs of lines. So, the far right column of the SSDs in Fig. 2A is the intersection of subsets all with members, the far left column is the intersections of all empty subsets, and the fourth from the left is $\neg A \cap B \cap C$, for instance. The construction of the SSDs is systematic and straightforward: for each non empty minimal subset draw a column with thick lines for sets that include members in that subset. Alternatively, we could draw Fig. 2A and just delete those subsets that are not in the given list. Fig. 2B shows a SSD in which one set does not intersect any of the others. Fig. 2C shows that SSDs can be drawn with a distinctive pattern for nested subsets, like Euler Diagrams. Fig. 2D shows that the number of segments for each set in a SSD equals the number of zones encompassed by the circle for the set in the Euler diagram (A=2, B=4, C=3). In some versions of Euler diagrams shading is used to show empty zones, such as Fig. 2E, in order to avoid complex constructions. However, the format and drawing method of SSDs does not change for any of the 218 diagrams that are configurations of three sets. Drawing

all those diagrams is a simple but tedious task with SSDs, which can be readily automated (e.g., using a spread sheet), but generating well-formed Euler diagrams is sufficiently demanding to have warranted a line of research in its own right [9].

Although, empirical studies will be required to test whether SSDs are easier to interpret than, say, Euler diagrams, the examples show that they are not particularly complex to draw. Thus, SSDs could potentially serve as an underpinning representation for knowledge rich domains.

4 PSD: Probability puzzle

The benefits of the PSD encoding of probability theory, in comparison to conventional notations, are described elsewhere and include empirical evaluations that show they can significantly improve students' comprehension and ease of learning [2]. A particularly compelling argument for the efficacy of PSDs is their ability to explain notorious probability puzzles and paradoxes, such as the Monty Hall Dilemma or Simpson's paradox [6]. So, let us consider the *Two Envelopes Problem* [16].

In this puzzle we consider two identical envelopes: we are told that one contains twice the amount of money than the other. The puzzle argues that it is always better to pick one, any one, then swap to the other envelop in order to maximize your chances of winning a larger amount on average, even if your initial choice of envelop is random. The argument goes like this: You pick one envelope and let's suppose that it actually contains $20. You swap, so there is an equal chance that you will pick the envelop with double the money, $40, or half the money, $10, so the expectation for swapping is $40*0.5+$10*0.5=$25, which is greater than $20 for sticking with your first choice! The outcome of argument is the same even if other amounts are chosen. The conclusion is clearly false, but why?

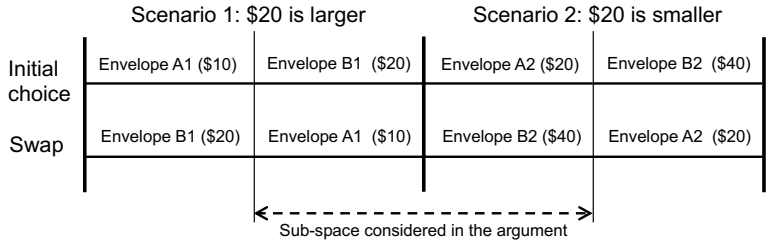

Fig. 3. PSD solution to the two envelop problem

Although the correct explanation is subject to debate [16], the PSD solution to the problem shows that the framing of a puzzle deliberately leads the problem solver into an elementary error of interpretation, Fig. 3. The argument assumes that the enveloped initially chosen contains $20 and other envelop could contain either a smaller or larger amount. These options are shown in the middle of the PSD. However, the argument ignores the possibilities that $20 might not have been the initial selection: the $10 option might have been chosen, left column; or $40 might have been chosen, right column.

The argument has selectively picked just one case from each scenario when there are, in fact, two cases. So, to compute the benefit of switching it is necessary so examine all four cases. Inspection of the top and bottom rows of Fig. 3 shows the amounts are equivalent, so on average there is no benefit of switching.

The PSD resolves the puzzle by encouraging the problem solver to model fully all of the possible outcomes. If we draw just the middle of Fig. 3 from the argument, our suspicion is immediately aroused that something is amiss, because of the lack of alternatives (both cases are $20) even though the problem talked about an initial random selection between two different options. This feature of PSDs is present, because it is based on SSDs, whose structure demands that all cases are enumerated in order to validly encode the underpinning set relations.

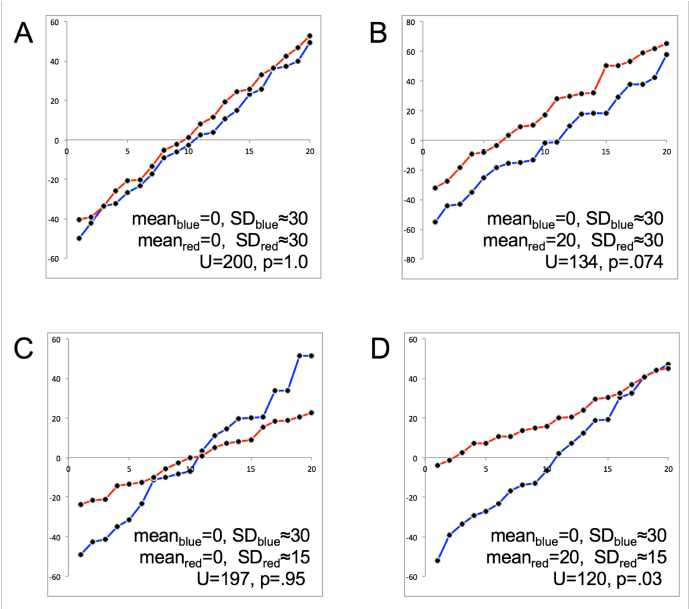

Fig. 4. Some DSD plots of pairs of independent data.

5 DSDs: Relations Between Distributions

The author has applied DSDs to a wide variety of problems involving statistical distributions, such as: visualizing large datasets; supporting the analysis of experimental results; interpreting statistical tests; explaining concepts that students find challenging to comprehend, such as the central limit theorem and the impact of distribution shape on statistical test validity. They will be described elsewhere: here we will use DSDs to examine relations between distributions.

Fig. 4 shows various pairs of datasets, *blue* and *red*, which happen to have the same number of data points but are sampled from unrelated measures (not matched). The data points (elements) are arranged in order of their magnitudes, with equal horizontal

spacing. Plotting multiple samples in DSDs allows us to easily compare distributions, because the shape of DSD data curves provides useful information. It is obvious that all the samples in Fig. 4 have uniform distributions as all their points have a largely linear arrangement. The amount of noise associated with each one is approximately equal; imagine a best fit straight line for each curve, the deviations of points from the line are similar for each sample. The difference between successive values is similar in all the samples in Figs. 4A and 4B, and for all the blue data in Fig. 4C and D. But the difference between values in the red datasets in Fig. 4C and D is about half the others.

We can also see that there is little overall difference between the red and blue samples in Fig. 4A, but there is a 20 unit difference in Fig. 4B. To judge whether this is likely to be a statistically significant difference, we can estimate what proportion of data points in one distribution is greater (or less) than the data points in the other distribution. Heuristically, this is akin to finding sums of ranks in a non-parametric Mann-Whitney U test [15]. In Fig. 4A, the first two blue data points are less than all the red values, but as we consider successive data points the number blue and red data points below a given value are nearly equal. This is true across most of the range of the distributions. In contrast, in Fig. 4B, consider the number of points below a given magnitude, there are always more blue points than red (until the very top of the distributions), so the apparent systematic difference between the samples is relatively unlikely to be due to chance, consistent with the outcomes of the tests shown in Fig. 4. Although, the proportion of red to blue points changes over the range of distributions Fig. 4C, it is obvious from the central intersection and the symmetry of the two curves that neither the red nor the blue points will overall tend to be larger than the other. Fig. 4D is interesting as the difference in means is the same as in Fig. 4B, but the reduced dispersion of the red data increases the overall difference, and hence it is most likely that the blue and red points come from different populations.

DSD curves for normal distributions are more complex (Fig. 1CiiB) than the linear forms in Fig. 4, but similar inferences are nevertheless easily made with normal curves. Further, an argument can be made that comparisons between normal distributions is easier with DSDs than probability density functions, because: (i) DSD normal curves have just two points of inflection, but bell curves have three; and (ii) probabilities are simply lengths in DSDs, whereas areas under non-linear segments of curves must be computed in probability density function plots. Moreover, the combination of the relative graphical simplicity and good expressive power mean that DSDs can perform the functions of myriad visualizations that are currently used to analyze distributions.

6 Discussion

The three specific examples presented above are suggestive of the potential of SSDs, PSDs and DSDs for supporting comprehension and problem solving in their respective domains. Like the author, readers may test the benefits and limitations of the representations by applying them to a variety of tasks and using them to recode information presented by other notations and visualizations.

DSDs use PSDs as a foundation, and in turn PSDs use SSDs as a foundation, so ideas about sets permeates the whole DS Framework. The ready access that these representations give to information about partitions and collections of elements, and relations between groups of elements, underpins the apparent benefits of PSDs and DSDs. For the probability puzzle the incompleteness of the space of options was made clear by drawing the PSD. For the comparison of distributions within and between the DSDs, in Fig. 4, inferences were supported both by the space for values (vertical dimension) and by the space of likelihoods of selected groups of data points (horizontal dimension). As such, PSDs and DSDs support the claim made in the introduction about the design of representations for knowledge rich domains: systematically capturing information about sets and closely integrating it with the other information from the domain can be a productive way to design representations that are generally effective. Knowledge about sets is fundamental, so starting with sets may be a good starting point for design.

The key to the successful design of the three representations resides in the simplicity of SSDs, which primarily used horizontal spatial arrangements of line segments to encode set relations. This left open the metric interpretation of lines to encode measures of chance in PSDs, and further left open the use of vertical space for magnitudes of domain properties and horizontal positioning for ranking of domain properties in DSDs; all without any substantive changes to the basic syntax and semantics of SSDs.

To finish, here are some open questions. Could representations for sets based on degrees of spatial containment, such as Euler diagrams, be used as coherent foundational representations for another encoding of probability and statistical distributions? How would it compare to the representations in the DS Framework? Could SSDs provide a foundational representation for other knowledge rich domains, such as timetabling, production planning and scheduling, or system specification, for which notations based on Euler diagrams have already been devised [4,5,12]?

Acknowledgement

This work was supported by EPSRC grant EP/R030642/1.

References

1. Cheng, P. C.-H. (2002). Electrifying diagrams for learning: principles for effective representational systems. Cognitive Science, 26(6), 685-736.
2. Cheng, P. C.-H. (2011). Probably good diagrams for learning: Representational epistemic re-codification of probability theory Topics in Cognitive Science, 3(3), 475-498.
3. Cheng, P. C.-H. (2014). Graphical notations for syllogisms: How alterative representations impact the accessibility of concepts. Journal of Visual Languages and Computing, 25, 170-185.
4. Cheng, P. C.-H., Barone, R., Cowling, P. I., & Ahmadi, S. (2002). Opening the information bottleneck in complex scheduling problems with a novel representation: STARK diagrams. In M. Hegarty, B. Meyer, & N. H. Narayanan (Eds.), Diagrammatic representations and inference: Second International Conference, Diagrams 2002 (pp. 264-278). Berlin: Springer-Verlag.

5. Cheng, P. C.-H., Barone, R., Pappa, N., Wilson, J. R., Cauvain, S. P., & Young, L. S. (2006). Understanding bakery scheduling: diverse methods for convergent constraints in user-centred design. In P. D. Bust (Ed.), Contemporary Ergonomics (pp. 45-49). London: Taylor & Francis.
6. Cheng, P. C.-H., & Pitt, N. G. (2003). Diagrams for difficult problems in probability. Mathematical Gazette, 87(508), 86-97.
7. Englebretsen, G. (1992). Linear diagrams for syllogisms (with relations). Notre Dame Journal of Formal Logic, 33, 37-69.
8. Euler, L. (1843). Lettres a une princesse d'Allemagne sur divers sujets de physique et de philosophie (1761). Charpentier.
9. Flower, J., & Howse, J. (2002). Generating Euler diagrams Diagrammatic Representation and Inference (pp. 61-75). Berlin Heidelberg: Springer.
10. Gottfried, B. (2014). Set space diagrams. Journal of Visual Languages and Computing, 25(4), 518-532.
11. Gottfried, B. (2015). A comparative study of linear and region based diagrams. Journal of Spatial Information Science, 10, 3-20. doi:10.5311/JOSIS.2015.10.187
12. Howse, J., Stapleton, G., & Taylor, J. (2005). Spider Diagrams. LMS Journal of Computation and Mathematics, 8, 145-194.
13. Jones, S. (1998). Graphical query specification and dynamic result previews for a digital library Proceedings of the 11th annual ACM symposium on User interface software and technology (pp. 143-151). San Francisco, CA: ACM.
14. Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Cognitive Psychology, 17, 248-294.
15. Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. Annals of mathematical statistics, 18(1), 50-60.
16. Nalebuff, B. (1989). Puzzles: the other person's envelope is always greener. Journal of Economic Perspective, 3(1), 171-181.
17. Spiegelhalter, D., Pearson, M., & Short, I. (2011). Visualizing uncertainly about the future. Science, 333, 1393-1400.
18. Venn, J. (1971). *Symbolic Logic, 2nd ed.* New York: Burt Franklin.
19. Zhang, J. (1997). The nature of external representations in problem solving. Cognitive Science, 21(2), 179-217.